Technical Perspective: Efficient Query Processing for Dynamically Changing Datasets

Wim Martens
University of Bayreuth, Germany

The paper *Efficient Query Processing for Dynamically Changing Datasets*, by Muhammad Idris, Martín Ugarte, Stijn Vansummeren, Hannes Voigt, and Wolfgang Lehner studies two central aspects of answering queries: (1) enumerating the answers to a query and (2) changing data. It is based on two papers by the same authors or a subset thereof, namely *The Dynamic Yannakakis Algorithm: Compact and Efficient Query Processing Under Updates* [4] and *Conjunctive Queries with Inequalities Under Updates* [5].

In a nutshell, these papers show how theoretical ideas for enumerating the answers to a query (e.g., in [1]) can be brought to the point where they actually work in practice and, furthermore, can deal with updates to the data. The fact that this was possible was not at all clear from the original theoretical work, which makes the current work extremely valuable to our community. Idris et al. show in their experiments that their algorithms perform consistently better than competitor incremental view maintenance systems with up to two orders of magnitude improvements in both time and memory consumption. They also show how the algorithms can be extended to deal with more general join conditions. So, they don't just work, they actually work *really well*.

In a classic paper from 2007, Bagan et al. studied for which conjunctive queries it is possible to enumerate the answers in *constant delay*, after *linear precomputation*. This means that an algorithm is first allowed to spend linear time in the database for computing a data structure, from which it is then possible to generate the answers of the query such that the time interval between consecutive answers does not depend on the size of the data. The entire complexity analysis is done in *data complexity*, which means that it only takes the size D of the database into account and considers the size Q of the query to be a constant. This means, more concretely, that a run-time of $2^{O(Q)} \cdot O(D)$ would be considered to be linear in the database — this fact may clarify to the attentive reader why bringing such an approach to practice may indeed be challenging. A main contribution of Bagan et al. is the result that says that, if acyclic queries are *free-connex*, then they can be evaluated with linear precomputation and constant delay. However, if they are not (and fulfill mild additional constraints [1]), then they cannot be evaluated within this time bound (under the conjecture that Boolean $n \times n$ matrices cannot be multiplied in time $O(n^2)$).

Idris et al.'s work provides an algorithm that computes a data structure that depends on the data and the query, and supports constant-delay enumeration for answering the query. The algorithm does not only support (projection-free) join queries, but also free-connex acyclic conjunctive queries which is, by results of Bagan et al. [1] and Brault-Baron [3], the largest class of conjunctive queries for which such an algorithm is possible under complexity-theoretical assumptions. The approach is heavily based on Yannakakis' algorithm for acyclic conjunctive query evaluation [6].

The approach is also robust under updates in the sense that tuple insertions or deletions can be propagated efficiently into the data structure. For so-called *q-hierarchical* queries, it is able to deliver (1) constant-delay enumeration of query results and (2) update propagation in time linear in the size of the update. Berkholz et al. [2] proved that the q-hierarchical queries are precisely the conjunctive queries that allow such an algorithm, unless the Online Matrix-Vector Multiplication conjecture is false.

This is a paper written by researchers with solid backgrounds in systems and theory *and it shows*. It provides algorithms for query evaluation that match rather tightly with theoretical lower bounds and perform very well in the experimental settings.

1. REFERENCES

[1] G. Bagan, A. Durand, and E. Grandjean. On acyclic conjunctive queries and constant delay enumeration. In *Computer Science Logic (CSL)*, pages 208–222, 2007.
[2] C. Berkholz, J. Keppeler, and N. Schweikardt. Answering conjunctive queries under updates. In *Symposium on Principles of Database Systems (PODS)*, pages 303–318, 2017.
[3] J. Brault-Baron. *De la pertinence de l'énumération: complexité en logiques propositionelle et du premier ordre*. PhD thesis, Université de Caen, 2013.
[4] M. Idris, M. Ugarte, and S. Vansummeren. The dynamic Yannakakis algorithm: Compact and efficient query processing under updates. In *International Conference on Management of Data (SIGMOD)*, pages 1259–1274, 2017.
[5] M. Idris, M. Ugarte, S. Vansummeren, H. Voigt, and W. Lehner. Conjunctive queries with inequalities under updates. *PVLDB*, 11(7):733–745, 2018.
[6] M. Yannakakis. Algorithms for acyclic database schemes. In *International Conference on Very Large Data Bases*, pages 82–94, 1981.